# EXHIBIT 42

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN JOSE DIVISION

| | |
|---|---|
| CISCO SYSTEMS, INC., Plaintiff, vs. ARISTA NETWORKS, INC., Defendant. | Case No. 5:14-cv-05344-BLF (PSG) |

VIDEOTAPED DEPOSITION OF GREG SATZ
March 23, 2016
HIGHLY CONFIDENTIAL - ATTORNEYS' EYES ONLY
VOLUME 1

Reported by
Brooke R. Bohr
CSR No. 753
Job No 2272380
Pages 1 - 168

point it out, there's two that I used. That's funny.

Q. Well, I believe later there -- it -- I think it was later, there was something called radius dash server?

A. It was a follow-on to the TACACS protocol.

Q. Okay.

A. That was the next generation of the solution to authenticate.

Q. And where did radius come from? Do you know?

A. Part of the protocol standard of whoever they called it. Again, it is name-calling. Somebody comes up with a name. It is like any Microsoft or somebody code names Apple, right, with their iPhone or their car products. It is secret names.

Q. Well, do you know where the radius protocol --

A. I don't recall anymore.

Q. -- where it came from?

A. I think I implemented that one, too. Yeah, I think the reason I used server is I equated them to Unix services as an optional

service. And so I called them a server, like you would on an operating system running as a separate process. So it was just a distinction I happened to use just from where I had come from.

Q. In the Unix context, what -- how was that manifest itself, or what's an example of that in the Unix context?

A. Well, Unix is what is on these phones. It is Linux. It is just the next generation of it. And so it is any arbitrary process running in the background that people might call a demon that provides a service. I mean you -- your -- you go to the web, you're talking to a web server. It just happens to be a dash in the configuration language.

Q. Right.

A. Maybe that helps with the modern analysis in comparison, as opposed to a routing protocol or a switching engine or a link layer, like an ARP. I mean, there's all these different components.

Q. Okay. I think we can put that aside.

So are you familiar with the terminal monitor command?

A. Yes.

Q. And do you know the origins of that?
A. I think I wrote it.
Q. Okay. What function does the terminal monitor command invoke?
A. I now use it without thinking. So the ability to figure out what's happening in a piece of software requires some diagnostics. And so we created a lot of debug commands that would print out the debugging. The debugging typically only went to the console which, in the good old big iron hardware, wasn't a bitmap display, but just an RS-232 port, and usually it was hooked to a good old-fashioned terminal in today's perspective. So the stream of debug diagnostic messages would come out this console port and if you're sitting at home, trying to connect in and do some debugging, it couldn't get there from here. The data was going over to your office in some terminal and the only way to look at it was to attach a stream back to where you were, and that was what monitor did. It said send me anything that came out on the console to my virtual terminal connection.
Now all laptops have bitmap displays and fancy graphics and no one even says RS-232

anymore.

Q. So am I right, then, that "monitor" in the command terminal monitor refers to monitoring the bug diagnostics?

A. It is actually monitoring anything that gets printed to the console port.

Q. Okay.

A. Which the important stuff was the diagnostics. It is the old world screen sharing of today.

Q. And did you write the code to implement that feature?

A. Yes.

Q. When did you do that, approximately?

A. Wherever it shows up in the manuals.

Q. Early, early years?

A. Yeah, because we needed that to help improve our proficiency to debug so we didn't have to be at the office.

Q. And how did you come upon the selection of the command terminal monitor for that?

A. The same expediency I did all of them: Monitor, sounds good, next.

Q. Okay.

A. Yeah. Unless Kirk didn't like my

choice, I think it was just whatever that struck me as a -- as what it did as I could perceive it from the point of view at the time.

Q. Have you ever heard of the term -- well, strike that.

Have you ever heard of people in your field characterizing a command as a "generic command"?

A. Yes.

Q. What does that mean to you?

A. Like "show." It is everywhere.

Q. And how would you contrast that concept of a generic command like show versus a non-generic command?

A. Its applicability to many different aspects or areas. So to -- I mean to use the word "generic" is not really clear, but it's -- probably if you look at it in the hierarchy sense, the top note is pretty generic. And depending on how many commands under it -- so relationship to the other commands around it or below it, if it is the root of a very deep tree, it's going to be more generic than if it's just one layer deep. And its applicability therefore expands out.

Q. So if I understand you, then, at a high

level something like the show family of commands is generic?

A. Set. I mean they are words that imply an action that can be used in a great many areas in the software.

Q. But you -- am I -- if I understand you, you would say something like show IPOSPF database, database dash summary might not be as generic?

A. The further down the tree you go, you're getting more specific.

Q. Okay. All right. Let's talk a little bit about Terry Slattery's work. And can you tell me, in general terms, what it was that Terry did that you hired Terry to do?

A. Terry's job was to take the Cisco source code and address how commands were parsed from the EXEC perspective that you used in the TOPS-20, which is the user interface commands which users type directly in the configuration commands, so that we would maximize code reusability and programmer efficiency.

So he and his team would generate a result that would improve our programming efficiency and maintenance overhead, to lower our maintenance overhead, which he did do.